

**A DYNAMICALLY PROGRAMMABLE INTEGRATED SWITCHING DEVICE  
USING AN ASYMMETRIC 5T1C CELL**

**FIELD OF INVENTION**

[0001] The present invention relates in general to electronic switching devices and elements and in particular to dynamically programmable integrated switching devices suitable for use in high speed routing and switching applications.

**BACKGROUND OF INVENTION**

[0002] In networked systems, the interconnect or the core switch fabric connecting the various system element, essentially attempts to connect  $N$  inputs to  $M$  outputs for the maximum number of possible routes. The "Non-Blocking" nature of the interconnection or the availability of "Clear Channels" enables the switch fabric to route or switch individual data packets.

[0003] In one interconnection architecture, the core switch fabric is based on time-domain multiple access (TDMA) to a common backplane or a shared bus. A controller, together with software, acts as the bus master and implements the routing kernel. The routing kernel is usually implemented in an algorithm such as a Hierarchical Weighted Fair Queuing algorithm.

[0004] Alternatively, the core switch fabric may be based on single or multiple crossbar integrated circuits. In this case, the controller asserts appropriate read and write commands to the crossbar and controls the exchange of data with a set of input and output buffers, typically constructed from common memory elements such as DRAM and SRAM. Switches are then built by using

multiple cards which connect to the multiple input and output ports of the crossbar with a non-blocking switch fabric.

[0005] In any event, the devices interfacing with the switch fabric are reaching higher and higher speeds. This in turn requires higher throughput rate through the switch fabric itself. Existing systems calculate aggregate throughput, in bits per second, by taking the throughput in bits per second for one port of the switch fabric and multiplying it by the total number of input and output ports. This aggregate capacity can be increased by varying the number of input and output ports on the switch fabric, the speed of operation of the switch fabric and the efficiency of the network processor. Notwithstanding, device physics and the electrical characteristics of busses and interconnects are still significant limiting factors on throughput speed.

[0006] Consequently, a switch element is required, which taken individually or in conjunction with other elements of a similar type, enables the design and fabrication of high speed scalable switch fabrics .

## **SUMMARY OF INVENTION**

[0007] According to one embodiment of the principles of the present invention, a switching element is disclosed which includes first, second and third ports each comprising a plurality of lines. A first memory cell includes a storage element, a first pass gate for selectively coupling a first line of the first port to the storage element, a second pass gate for selectively coupling a first line of the second port to the storage element, and a third pass gate for selectively coupling a first line of the third port to the storage element. The switching element also includes a second memory cell having a first pass gate for selectively coupling a second line of the first port to the storage element, a second pass gate for

selectively coupling a second line of the second port to the storage element, and a third pass gate for selectively coupling a second line of the third port to the storage element.

[0008] Switching elements, switches and switching subsystems embodying the principles of the present invention enable the design and fabrication of high speed scalable switch fabrics. Such high-speed switch fabrics are particularly useful in network switches and routers, although not necessarily limited thereto.

#### **BRIEF DESCRIPTION OF DRAWINGS**

- [0009] FIGURE 1A is conceptual block diagram of a router or a switch;
- [0010] FIGURE 1B is the logical block diagram of a switch router with input and output queues;
- [0011] FIGURE 2A is the functional block diagram of a router designed with forwarding engines;
- [0012] FIGURE 2B is the functional block diagram of a router designed with Interfaces and a core switch fabric;
- [0013] FIGURE 3 is the general architectural block diagram of a typical Input Output Interface;
- [0014] FIGURE 4A is the general logical block diagram of a Broadcast Switch Element (BSE);
- [0015] FIGURE 4B is the general logical block diagram of a Receive Switch Element (RSE);

[0016] FIGURE 5A is the circuitry diagram of a Broadcast Switch Element implemented using a 5T1C cell;

[0017] FIGURE 5B is the timing diagram of a read-write cycle for a BSE;

[0018] FIGURE 6A is the circuitry diagram of a Receive Switch Element implemented using a 5T1C cell;

[0019] FIGURE 6B is the timing diagram of a read write cycle for a RSE;

[0020] FIGURE 7 is the circuitry diagram of a port block formed by a 5T1C BSE;

[0021] FIGURE 8 is the functional block diagram of a port block formed by 5T1C BSE;

[0022] FIGURE 9 is the architecture of a switching device formed with by port blocks implemented with 5T1C;

[0023] FIGURE 10 is the block diagram of a row within a switching device emphasizing the column decode;

[0024] FIGURE 11A is the functional block diagrams of the write decode block of FIGURE 9;

[0025] FIGURE 11B is a possible implementation of a decode from prior art; and

[0026] FIGURE 12 is the functional block diagram of a read decode block.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

[0027] A conceptual diagram of a switch /routing system architecture 100 is shown in Figures 1A and 1B. Switch fabric 101 in conjunction with the I/O of switch /router 100 can be visualized as a number of input and output queues 102, 103 by non-blocking interconnections 104. Interconnections 104 may be for example single or multiple stage crossbars or a backplane. The input and output queues 102, 103 are typically disposed on the I/O port cards 106a,f. System controller 105 implements a queuing/ de-queuing algorithm (kernel), and generally controls the core switch fabric under software and firmware control.

[0028] Exemplary router architectures based on the current generation of network processors are shown in FIGURE 2A and FIGURE 2B. In the system of FIGURE 2A, the processing power is in the hardware and software of forwarding engines 201. With respect to the system of FIGURE 2B, the processing power is in the systems interfaces 202, including the scheduling and system control functions. Specifically, the main difference between the architectures shown in Figures 2A and 2B is where the actual forwarding table resides (in FIGURE 2A in the forwarding engines and in FIGURE 2B in the system interfaces). These route tables can be represented by data structures generated by the network processor and are stored in the system memory.

[0029] With respect to FIGURE 3, a selected I/O interface 202 is modeled by the general structure shown. The forwarding engine is a firmware implementation of the algorithms. Memory buffers 301, 302 logically act as the input / output queues in the system. These memory buffers add delays to the whole process of taking a packet from the physical input port of the router (PHY Receive) to the physical output port (PHY Transmit) with the appropriate header / routing information.

[0030] FIGURE 4A depicts a Broadcast Switch Element, (BSMnm – taken from the nth row and mth column of the switch architecture discussed below). 401, logically represented by a 1 x K de-multiplexer having one input port (iBnm) and K output ports (0Bnmk). A Receive Switch Element (RSE) 402 is logically represented by a K x 1 multiplexer in FIGURE 4B and has K input (iRnmk) ports and one output port (0Rnm).

[0031] According to the principles of the present invention, a 1 x 4 BSE 401 is implemented by a 5T1C (5 transistor, 1 capacitor) dynamic memory cell shown in FIGURE 5. The input port (gate) 501, labeled iBnm, and output ports 502a,d, labeled 0Bnm1 to 0Bnm4 are formed by metal oxide semiconductor field effect transistors (MOSFETs). Specifically, the first output port is formed by the output transistor 502a, the second output port is formed by the transistor 502b, the third output port is formed by the transistor 502c and the final and fourth output port is formed by the transistor 502d. Each 5T1C cell has a single storage element represented by the capacitor 503.

[0032] Exemplary read and write cycles for BSE 401 element are shown in FIGURE 5B, where the appropriate gates are turned-on as indicated by the assertion of the read and write enables. In the write cycle, the input gate 501 is turned-on with the signal WRITE ENABLE WE and the storage capacitor 503 is allowed to charge to a level proportionate to the input gate transistor drive. The voltage across the storage capacitor is a function of the current and the charging time is dictated by the time constant.

[0033] Data written into the storage capacitor can be read out by selectively turning on the output transistors 502a,d either individually, all at once, or in some other combination, by selecting the corresponding READ ENABLE signal RE1-RE4. In particular, if the port block, described below, to which the

specific BSE belongs, is being employed for a multicast session then all the output gates can be turned on simultaneously. Otherwise the gates are normally turned on individually. To read from the storage element simultaneously with a write, a feedback mechanism external to the basic switch element retains the data and writes them back into the storage capacitor 503 in an off cycle.

[0034] The inventive concepts can also be also be applied to RSE core 402 as shown in FIGURE 6A. Here, the RSE core is implemented with gate transistors 601a,d forming the input ports and the gate transistor 602 forming the output port. The storage element is again represented by a capacitor, in this case capacitor 603. It should be understood that at a given time during the operation of the RSE only one input port 601 may be used to write data into the storage element represented by the storage capacitor 603.

[0035] In case of an RSE, the operation is the reverse of the operation of the BSE, as shown in FIGURE 6B. In the first cycle, data can be written into storage capacitor 603, by the use of any one of the input port gates 601a,d and the WRITE ENABLE signals WE1-WE4. When multi-valued storage systems are possible using a single storage element, all four gates can be used concurrently to store multiple values into the storage capacitor 602. Data can be read from the output port gate 602 simultaneous with a write, if an external feedback mechanism is provided external to the core RSE switch element.

[0036] With respect to FIGURE 7, a port block 700 that is P bits wide is created using P number of 5T1C BSEs 500. All the input ports of the P number BSEs 500 are taken together to form the input port  $I_{PB}^{NM}$  of the port block 700 with each input controlled by a corresponding write enable signal WE1-WE4. Controlled by a corresponding write enable signal WE1-WE4. The illustrated port

block has 4 output ports  $O_{PB}^{NM1} - O_{PB}^{NM4}$ . The first output line of the first output of each BSE 500 are tied together to form the output port 1  $O_{PB}^{NM1}$ . In a similar fashion, output port 2  $O_{PB}^{NM2}$  of the port block is formed by taking all the second output ports of each of the BSE 500 together, and so on such that, each of the output ports are formed in a linear fashion. Other nonlinear combination of BSE can be used to form a port block. FIGURE 8 shows the interface diagram of the port block.

[0037] A switch matrix of size  $N \times M$ , within the DIPS device (900), is formed by port blocks 700 arranged in rows and columns as shown in FIGURE 9. (It is not necessary that the individual port blocks are arranged in a row column fashion and interconnected in a matrix format.) In addition to the matrix of port blocks 700, DIPs device 900 also includes Write Decode and Read Decode blocks 901, 902, Lookup Decode 903 and controls 904.

[0038] With respect to FIGURE 10, each row  $N$  of port blocks has one  $P$ -bit wide input  $I_N$ , this input feeds into a 1 to  $M$  input demux (1001). This de-mux is a form of decode and essentially is part of the write decode block 901. Demux 1001 is preferably of a conventional design, using combinational circuits such as cascaded, domino etc. Based on the decode code given to the decode circuit, the input data on the input port  $I_N$  is sent to the appropriate port block in the row. For each row  $N$  in the DIPS device there is one input de-mux 1001, that allows one input to be tied to each of the inputs of the  $M$  port blocks in a row.

[0039] When each port block comprises 4 5TIC memory cells, each row of port blocks 700 has four outputs  $O_{N1} - O_{N4}$  that are each  $P$  bits wide, each coupled through an output mux (1002). Each output mux (1002) is a  $M$  to 1 mux. Preferably, each of the  $P$ -bit wide outputs of the port blocks are tied to the output

muxes 1002 as follows; the first output  $O_{PB}^{NM1}$  of each of the M port blocks 700 in the row, first output mux 1002a, the second output  $O_{PB}^{NM2}$  of each port 700 blocks is an input to the second output mux 1002b and so on for all the four outputs.

[0040] Output muxes 1002 are part of read decode block 902 in the DIPS device 900. Each of output muxes are formed by combinatorial circuits and implement a 1 of M decode. The outputs of each of these muxes are sent to an I/O block that is part of the controller (904) for the DIPS device. DIPS device 900 has a single output through the output port of the device which is P bits wide. DIPS device 900 also includes a single input port that is also P bits wide. These constraints are placed on the DIPS device due to semiconductor packaging limitations.

[0041] FIGURES 11A and 11B are more detailed diagrams of Write Decode block 901. The output of input mux 1001 is sent to a write drive block (1101) that ties into the input gate of each BSE. FIGURE 12 is a more detailed diagram of Read Decode block 902. Each of the output gates of the BSE tie into an amplification block (1201), that is formed by a differential amplifier as shown. The outputs of the differential amplifier drive the inputs to the combinatorial output mux 1002. Within the port block, a reference cell can be used to drive the differential inputs to the amplifier 1201 or a shadow, 5T1C cell that is used for redundancy can be used, to drive the reference input to the differential amplifier.

[0042] If the shadow 5T1C cell is used then each of the port blocks forms a mirrored memory element and switch. The use of the mirrored memory element and switching device can be used to control errors in reading or writing. This implements a pseudo cache.

[0043] With respect to FIGURE 12, the write decode block is implemented to form a 1 to M decode for each port block. A control input that is  $\text{Log}_2 (NM)$  bits wide is decoded into the appropriate port block address within the row. A simple decode scheme is shown in this embodiment. It should be clear to those of ordinary skill in the art, decode can be changed without departing from the spirit of the invention.

[0044] The operation of DIPS 900 device can be summarized as follows:

[0045] 1) An external Switch Controller asserts the appropriate read and write signals to the DIPS device that is part of the Switch fabric matrix.

[0046] 2) The reads and write signals are decoded for the assertion of the reads and writes to the port blocks internally within the DIPS (900) device by the controll (904).

[0047] 3) The reads and writes are decoded by the read-decode blocks and the write-decode blocks within the DIPS device.

[0048] 4) The write and reads are done asynchronously and in the same clock cycle, thus in a given clock cycle at the minimum, using a simple linear decode one can access two port blocks.

[0049] The throughput thus of a DIPS device based on the aforementioned protocol followed by the read and write cycles, is  $2 * P\text{bits} * \text{Speed in Mhz}$  of the DIPS device. Thus for a 100 Mhz DIPS device with a port block that is 64 bits wide the throughput of a DIPS device is  $= 2 * 64 * 100 \text{ Mhz} = 12.8 \text{ Gbps}$  for a DIPS device. For a fabric implemented by using multiple DIPS devices throughput is  $\# \text{ DIPS device} * 12.8 \text{ Gbps per DIPS device}$ .

ATTORNEY DOCKET NO.  
24155- P002US

PATENT

[0050] A similar implementation of the DIPS device can be done using the RSE. While a particular embodiment of the invention has been shown and described, changes and modifications may be made therein without departing from the invention in its broader aspects, and, therefore, the aim in the appended claims is to cover all such changes and modifications as fall within the true spirit and scope of the invention.